COMPUTER AIDED DESIGN SYSTEMS AND METHODS WITH REDUCED MEMORY UTILIZATION

RELATED APPLICATIONS

[0001] The present document contains material related to the material of copending, cofiled, U.S. patent applications Attorney Docket Number 100111221-1, entitled System And Method For Determining Wire Capacitance For A VLSI Circuit; Attorney Docket Number 100111227-1, entitled System And Method For Determining Applicable Configuration Information For Use In Analysis Of A Computer Aided Design; Attorney Docket Number 100111228-1, entitled Systems And Methods Utilizing Fast Analysis Information During Detailed Analysis Of A Circuit Design; Attorney Docket Number 100111230-1, entitled Systems And Methods For Determining Activity Factors Of A Circuit Design; Attorney Docket Number 100111232-1, entitled System And Method For Determining A Highest Level Signal Name In A Hierarchical VLSI Design; Attorney Docket Number 100111233-1, entitled System And Method For Determining Connectivity Of Nets In A Hierarchical Circuit Design; Attorney Docket Number 100111234-1, entitled System And Method Analyzing Design Elements In Computer Aided Design Tools; Attorney Docket Number 100111235-1, entitled System And Method For Determining Unmatched Design Elements In A Computer-Automated Design; Attorney Docket Number 100111238-1, entitled System And Method For Iteratively Traversing A Hierarchical Circuit Design; Attorney Docket Number 100111257-1, entitled Systems And Methods For Establishing Data Model Consistency Of Computer Aided Design Tools; Attorney Docket Number 100111259-1, entitled Systems And Methods For Identifying Data Sources Associated With A Circuit Design; and Attorney Docket Number 100111260-1, entitled Systems And Methods For Performing Circuit Analysis On A Circuit Design, the disclosures of which are hereby incorporated herein by reference.

BACKGROUND

[0002] An electronic computer aided design ("E-CAD") package is utilized to construct a Very Large Scale Integration ("VLSI") circuit design. The VLSI circuit design consists of a netlist that identifies electronic design elements (e.g., capacitors, transistors, resistors, etc.) and their interconnectivity (e.g., signal nets) within the VLSI circuit design. The VLSI circuit design is constructed from hierarchical design blocks (also known as cells) that provide specific functionality to the VLSI circuit design. Such design blocks may be re-used within the VLSI circuit design, or within other circuit designs. Designs blocks may be constructed from electronic design elements, nets and other design blocks, and may be re-used one or more times. Each use of a design block is called an "instance."

[0003] A design engineer uses the E-CAD tool to analyze the VLSI circuit design during development. The E-CAD tool typically traces through instances of blocks used in the VLSI circuit design to sum certain information (e.g., field-effect transistor ("FET") width, wire capacitance, FET capacitance, etc.). During this analysis, the E-CAD tool typically loads the entire VLSI circuit design into memory. If the VLSI circuit design has billions of design elements, the circuit design uses large amounts of this memory. In certain situations, processing the circuit design becomes limited by the amount of available memory, restricting development and slowing production. Continuous lost productivity due to lengthy engineering development slows technology advancement and can result in significant costs, as well as lost business.

SUMMARY OF THE INVENTION

[0004] In one embodiment, a method analyzes a circuit design with reduced memory utilization. Access to at least one block of the circuit design is detected. If the one block is not loaded within a circuit model of computer memory, a determination is made whether loading the one block into the circuit model would exceed a predefined maximum utilization of the computer memory. If loading the one block into the circuit model would exceed the predefined maximum utilization, one or more blocks from the circuit model are unloaded and the one block is loaded into the

circuit model. If loading the one block into the circuit model would not exceed the predefined maximum utilization, the one block is loaded into the circuit model.

[0005] In another embodiment, a system analyzes a circuit design with reduced memory utilization. A computer memory stores a circuit model of the circuit design. An analysis tool analyzes the circuit design by accessing one or more blocks of the circuit model. A model manager (a) loads one or more blocks of the circuit design to the circuit model and (b) unloads one or more blocks from the circuit model such that the circuit model does not exceed a predefined maximum utilization of the computer memory.

[0006] In another embodiment, a system analyzes a circuit design with reduced memory utilization, including: means for detecting access to at least one block of the circuit design; means for determining whether loading the one block into a circuit model, stored within computer memory, would exceed a predefined maximum utilization of the computer memory when the one block is not currently within the circuit model; means for unloading one or more blocks from the circuit model and loading the one block into the circuit model when loading the one block into computer memory would exceed the predefined maximum utilization; and means for loading the one block into the circuit model when loading the one block into the computer model would not exceed the predefined maximum utilization.

[0007] In one embodiment, a software product has instructions, stored on computer-readable media, wherein the instructions, when executed by a computer, perform steps for analyzing a circuit design with reduced memory utilization, including:

> instructions for detecting access to at least one block of the circuit design; instructions for recording the access when the one block is loaded within a circuit model of computer memory; instructions for determining whether loading the one block into the circuit model would exceed a predefined maximum utilization of the computer memory, when the one block is not loaded within the circuit model; instructions for unloading one or more blocks from the circuit model and loading the one block into the circuit model, when loading the one block into the circuit model would exceed the predefined maximum

utilization; and instructions for loading the one block into the circuit model, when loading the one block into the circuit model would not exceed the predefined maximum utilization.

BRIEF DESCRIPTION OF THE FIGURES

[0008] FIG. 1 shows an exemplary schematic diagram of one hierarchical circuit design.

[0009] FIG. 2 is a block diagram illustrating one computer aided design system with reduced memory utilization.

[0010] FIG. 3 is an exemplary schematic diagram illustrating five design blocks.

[0011] FIG. 4 illustrates exemplary hierarchical instances of the design blocks of FIG. 3.

[0012] FIG. 5 illustrates exemplary data flow during analysis of a circuit design by a computer aided design system with reduced memory utilization.

[0013] FIG. 6 shows three chronological snapshots of a model manager.

[0014] FIG. 7 is a flowchart illustrating one process for updating a circuit model to reduce memory utilization.

[0015] FIG. 8 shows one method for analyzing a circuit design with reduced memory utilization.

DETAILED DESCRIPTION OF THE FIGURES

[0016] A signal net is a single electrical path in a circuit design that has the same electrical characteristics at all of its points. Any collection of wires that carries the same signal between design elements is a signal net. If the design elements allow the signal to pass through unaltered (as in the case of a terminal), then the signal net continues on subsequently connected wires. If, however, the design element modifies the signal (as in the case of a transistor or logic gate), then the signal net terminates at that design element and a new signal net begins on the other side. Connectivity in a circuit design is typically specified using a netlist, which indicates the specific nets that interconnect the various design elements.

[0017] A signal net may be divided into signal net 'pieces', each of which is part of a Highest Level Signal Name ("HLSN"). A HLSN is the unique signal name

that identifies a collection of signal nets or 'hierarchical signal net pieces', which are the small pieces of intervening wire (signal nets) in each hierarchical design block of a circuit design.

[0018] A significant characteristic of VLSI and other types of circuit design is a reliance on hierarchical description. A primary reason for using hierarchical description is to hide the vast amount of detail in a design. By reducing the distracting detail to a single object that is lower in the hierarchy, one can greatly simplify many E-CAD operations. For example, simulation, verification, design-rule checking, and layout constraints can all benefit from hierarchical representation, which makes them more computationally tractable. Since many circuit designs are too complicated to be easily considered in their totality, a complete design is often viewed as a collection of design element aggregates that are further divided into sub-aggregates in a recursive and hierarchical manner. In VLSI circuit design, these aggregates are commonly referred to as design blocks (or cells). The use of a design block at a given level of hierarchy is called an 'instance'. Each design block has one or more 'ports', each of which provides a connection point between a signal net within the design block and a signal net external to the design block.

[0019] FIG. 1 illustratively shows one hierarchical circuit design 10. Circuit design 10 has a block instance 12 that includes two block instances 14 and 14' of a design block. Block instance 14 has four ports 28, 30, 32 and 34 and, in this example, includes a p-type field-effect transistor ("FET") 16 and an n-type FET 18 connected to form an inverter. Block instance 14 also has four signal nets: 20, 22, 24 and 26: signal net 20 connects to port 32; signal net 22 connects to port 34; signal net 24 connects to port 28; and signal net 26 connects to port 30. Block instance 14' is similarly shown with FETs 16' and 18', signal nets 20', 22', 24', and 26', and ports 28', 30', 32', and 34'.

[0020] To illustrate exemplary nomenclature used in analyzing block instance 12, block instance 12 is shown with five signal nets: input net 36, pass net 38, output net 40, VDD net 42 and GND net 44. Signal nets 20 and 36 connect to port 32, forming hierarchical signal net pieces identified by HLSN "input net". Signal nets 22, 38 and 20' interconnect by ports 34 and 32' and form hierarchical signal net pieces identified by HLSN "pass net". Signal nets 24, 42 and 24' interconnect by

ports 28 and 28' and form hierarchical signal net pieces identified by HLSN "VDD net". Signal nets 26, 44 and 26' interconnect by ports 30 and 30' and form hierarchical signal net pieces identified by HLSN "GND net". Signal nets 22' and 40 connect to port 34' and form hierarchical signal net pieces identified by HLSN "output net". Block instance 12 further includes ports 46, 48, 50 and 52 that connect internal signal nets 36, 42, 40 and 44 to signal nets external to block instance 12.

[0021] FIG. 2 is a block diagram illustrating one computer aided design system 100 with reduced memory utilization. System 100 has a computer 102 with a computer memory 104, a processor 106, a storage unit 108 and a user interface 110. Storage unit 108 may for example be a disk drive for storing programs and data of computer 102. Storage unit 108 is illustratively shown storing an E-CAD tool 114, an analysis tool 120, and a circuit design 116. Circuit design 116 is, for example, a circuit design created by E-CAD tool 114. Circuit design 116 includes a netlist 118 that defines signal nets interconnecting various design elements of circuit design 116. E-CAD tool 114 includes a model manager 122 operable to reduce memory utilization of E-CAD tool 114 by partially loading circuit design 116 into a circuit model 124 for analysis tool 120.

[0022] More particularly, processor 106 loads E-CAD tool 114, including model manager 122 and analysis tool 120, from storage unit 108 into computer memory 104 such that E-CAD tool 114 and model manager 122 are executable by processor 106. Once loaded into computer memory 104, a design engineer operates E-CAD tool 114 to process and analyze circuit design 116. During analysis of circuit design 116 by analysis tool 120, model manager 122 operates to reduce memory utilization by analysis tool 120; model manager 122 loads blocks of circuit design 116 into circuit model 124 (within computer memory 104), as required by analysis tool 120, and unloads blocks from computer memory 104 to storage unit 103 when unnecessary to analysis tool 120.

[0023] In illustrative operation, user interface 110 connects to a terminal 112 (e.g., a keyboard), external to computer 102. Through terminal 112 and user interface 110, the design engineer interacts with E-CAD tool 114 and analysis tool 120. In one example, the design engineer instructs E-CAD tool 114 to analyze circuit design 116 using analysis tool 120. Model manager 122 in turn manages which blocks

are present within computer memory 104 when analysis tool 120 analyzes circuit design 116.

[0024] An exemplary circuit design 116' is shown and described in connection with FIG. 3 and FIG. 4. Specifically, FIG. 3 shows five design blocks A, B, C, D and E of a hierarchical circuit design 116', which may for example represent circuit design 116 of FIG. 1. Similar to block 12, FIG. 1, which includes block instances 14 and 14', design block A includes design blocks B and C; design block B includes design block C; and design block C includes design blocks D and E. Design blocks D and E do not include other design blocks. Design elements are not shown within design blocks A, B, C, D and E of the example for clarity of illustration. A design engineer defines design blocks A-E prior to instantiation within circuit design 116'.

[0025] FIG. 4 is a block diagram illustrating exemplary hierarchical instances of design blocks A, B, C, D and E of circuit design 116'. Design block A is the 'top level block' of circuit design 116', and is instantiated as block instance 'A1', indicating that it is the first instance of block A. As design block A includes design blocks B and C, design blocks B and C are instantiated as block instances B1 and C1, as shown. Design block B includes design block C; thus a second instantiation of design block C is identified as block instance C2. Design block C includes design blocks D and E; a first instantiation of design blocks D and E is thus shown as block instances D1 and E1 associated with block instance C1; second instantiations of design blocks D and E are shown as block instances D2 and E2 associated with block instance C2. Accordingly, circuit design 116' has five design blocks A, B, C, D and E, each instantiated one or more times, and totaling eight instantiations A1, B1, C1, C2, D1, D2, E1 and E2.

[0026] By way of illustrative operation, and with further reference to FIG. 2, the design engineer instructs analysis tool 120 to perform circuit analysis on one or more selected blocks (A-E) of circuit design 116'. In this way, the design engineer can analyze entire circuit design 116' or only selected blocks of circuit design 116'. In one example, the design engineer instructs analysis tool 120 to determine power requirements of circuit design 116'. Analysis tool 120 may, for example, determine switching power requirements of all HLSN signal nets of circuit design 116',

hierarchically following the HLSN signal nets through selected blocks of circuit design 116'.

[0027] FIG. 5 is a block diagram illustrating exemplary data flow with reduced memory utilization during analysis of circuit design 116' by analysis tool 120. In this example, storage unit 108 is shown loaded with circuit design 116' of FIG. 4, and computer memory 104 is shown loaded E-CAD tool 114 and circuit model 124. E-CAD tool 114 includes model manager 122 and analysis tool 120. Model manager 122 has a block access table 126 to record access to blocks within circuit model 124, and a block threshold 128 that predefines a maximum number of blocks to be concurrently loaded in circuit model 124. Block threshold 128 therefore sets a predefined maximum utilization of computer memory 104; it may be set, for example, by a design engineer through user interface 110, FIG. 2. Model manager 122 manages circuit model 124 such that one or more portions of circuit design 116' are accessible by analysis tool 120; such portions are made accessible to analysis tool 122 by loading appropriate design blocks from storage unit 108 to computer memory 104. In the example shown in FIG. 5, analysis tool 120 has access to block instances C2, D2 and E2 of circuit model 124. Block access table 126 thus tracks access to blocks C2, D2 and E2 of circuit model 124 by analysis tool 120. Model manager 122 also uses information of block access table 126 to determine which blocks to unload from circuit model 124 to storage unit 108.

[0028] In one embodiment, for each detected access to each block within circuit model 124, the time of the access to each block (e.g., a timestamp) is stored in block access table 126; block access table 126 thereby records the time of the most recent access to each block in circuit model 124. In another embodiment, for each block loaded into circuit model 124, a timestamp is stored in block access table 126, indicating the block load time, and, for each detected access to blocks within circuit model 124, an associated count in block access table 126 is incremented; block access table 126 thereby stores the frequency of access to the block since the block was loaded into circuit model 126. Those of ordinary skill in the art appreciate that alternate information may be stored in block access table 126 without departing from the scope hereof.

[0029] To further illustrate management of circuit model 124 by model manager 122, FIG. 6 shows three chronologically sequenced snapshots (124(A), 124(B), 124(C)) of circuit model 124. In circuit model 124A, block instances C2, D2 and E2 are shown within circuit model 124 and available for access by analysis tool 120. Model manager 122 then determines that analysis tool 120 requires access to block instance B1. As model manager 122 has already loaded a maximum number of block into circuit design 124, in this example, model manager 122 utilizes block access table 126 to determine that block D2 was the least recently accessed block within circuit model 124 and should therefore be removed prior to loading block B1. Circuit model 124B thus shows block D2 removed from circuit model 124. Circuit model 124C similarly illustrates a third situation after model manager 122 loads block B1 from circuit design 116' into circuit model 124.

[0030] Model manager 122 may use any one or more known techniques to identify one or more blocks to be removed from circuit model 124. For example, least recently used ("LRU") and least frequently used ("LFU") methods may be used by model manager 122 to identify the blocks to be removed from memory 104. In one example, model manager 122 caches blocks of circuit design 116 in circuit model 124 such that analysis tool 120 is unaware of the reduced memory utilization.

[0031] FIG. 7 is a flowchart illustrating one process 700 for reducing memory utilization in analyzing a circuit design. In one example, process 700 is implemented by model manager 122, FIG. 2. In step 702, process 700 detects block access by analysis tool 120. For example, in step 702, process 700 detects access to a block (e.g., block C2) loaded within circuit model 124 or a request to access a block (e.g., block B1) not loaded into circuit model 124. Step 704 is a decision. If the access is to a block already loaded in circuit model 124, process 700 continues with step 716; otherwise, process 700 continues with step 706. Step 706 is a condition. If circuit model 124 contains the maximum amount of blocks, as defined by block threshold 128, process 700 continues with step 708; otherwise, process 700 continues with step 712. In step 708, process 700 determines which block of circuit model 124 to remove in order to load, within circuit model 124, the block identified in step 702. For example, in step 708, model manager utilizes block access table 126 when implementing caching techniques to identify the LRU block or LFU block within

circuit model 124. In step 710, the block identified in step 708 is removed, increasing available memory 104 by creating space within circuit model 124. In step 712, the block identified in step 702 is loaded from circuit design 116 into circuit model 124. In step 714, circuit model 124 is updated such that the block loaded in step 712 is integrated with other blocks within circuit model 124. In step 716, access to the blocks identified in steps 702 and 710 are recorded in block access table 126.

[0032] Those skilled in the art appreciate that process 700 may operate with similar function with reordered or modified steps without departing from the scope hereof.

[0033] FIG. 8 is a flowchart illustrating one process 800 for analyzing a circuit design with reduced memory utilization. In step 802, process 800 detects access to at least one block of the circuit design. In step 806, if the one block is not loaded within the circuit model, process 800 determines whether loading the one block would exceed a predefined maximum utilization of the computer memory. In step 808, if loading the one block into computer memory would exceed the predefined maximum utilization, process 800 unloads one or more blocks from the circuit model and loads the one block into the circuit model. In step 810, if loading the one block into computer memory would not exceed the predefined maximum utilization, process 800 loads the one block into the circuit model.

[0034] Upon reading and fully comprehending the above disclosure, one of ordinary skill in the art appreciates that by partially loading circuit design 116 into memory, E-CAD tool 114 operates with reduced memory utilization. This allows circuit designs larger than available memory to be developed and to utilize unused memory for other purposes.

[0035] Changes may be made in the above methods and systems without departing from the scope hereof. It should thus be noted that the matter contained in the above description or shown in the accompanying drawings should be interpreted as illustrative and not in a limiting sense. The following claims are intended to cover all generic and specific features described herein, as well as all statements of the scope of the present method and system, which, as a matter of language, might be said to fall there between.